

## darktable - Bug #11995

### EXIF information incomplete - camera recognized, lens not (old equipment)

02/03/2018 11:34 PM - Stefan Hoffmeister

<b>Status:</b>	Closed: invalid	<b>Start date:</b>	02/03/2018
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Darkroom	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.6.0	<b>bitness:</b>	64-bit
<b>Affected Version:</b>	2.4.1	<b>hardware architecture:</b>	amd64/x86
<b>System:</b>	Windows		

#### Description

The attached image has issues showing various pieces of EXIF information, although exiftool-10.49 (and probably later versions) are able to extract this information just fine.

In the "image information" section, darktable does not show (or does not show *correctly*)

- lens
- aperture
- exposure
- focal length
- focus distance
- ISO

EXIF tool shows, amongst other properties,

Make : OLYMPUS IMAGING CORP.

Camera Model Name : E-PL7

Exposure Time : 1/400

F Number : 10.0

ISO : 200

Focal Length : 7.0 mm

Lens Type : Lumix G Vario 7-14mm F4.0 Asph.

Lens Info : 7-14mm f/4

Lens Model : LUMIX G VARIO 7-14/F4.0

Focus Distance : inf

all of which appears to be correct. These pieces of equipment date from 2014 (camera) and 2010 (lens).

Note: It seems as if rendering of EXIF information is a bit weird: Only once an image has opened in "darkroom", will the camera model and maker information be updated. The remainder of the data is not rendered, ever, though.

#### History

##### #1 - 02/03/2018 11:35 PM - Stefan Hoffmeister

I cannot tell whether this is peculiar to darktable 2.4.1 on Windows, or whether this also crops up on Linux.

##### #2 - 02/03/2018 11:41 PM - Stefan Hoffmeister

This is on Windows 10 (current), with Swiss regional settings, but English language output. Right now, "." is my thousand separator, and "," my decimal separator. I think, based on what Windows Calculator wants me to use ...

##### #3 - 02/03/2018 11:51 PM - Stefan Hoffmeister

I just closed darktable and looked at USER\AppData\Local\darktablelibrary.db -> select filename, maker from images where (maker is not null) and (maker <> "")

It seems as if EXIF information is not located there at all, ever, not cached?

#### #4 - 02/03/2018 11:58 PM - Stefan Hoffmeister

Downloaded Windows binaries for exiv2 0.26 from <http://www.exiv2.org/download.html>, used the exiv2 command-line tool built by vs2015 for 64 bit and got the following (correct) output:

##### ORF IMAGE

File name : <removed>  
File size : 15143885 Bytes  
MIME type : image/x-olympus-orf  
Image size : 4640 x 3472  
Camera make : OLYMPUS IMAGING CORP.  
Camera model : E-PL7  
Image timestamp : 0000:00:00 00:00:00  
Image number :  
Exposure time : 1/400 s  
Aperture : F10  
Exposure bias : 0 EV  
Flash : No, auto  
Flash bias : 0/256  
Focal length : 7.0 mm  
Subject distance:  
ISO speed : 200  
Exposure mode : Auto  
Metering mode : Multi-segment  
Macro mode : Off  
Image quality :  
Exif Resolution : 4640 x 3472  
White balance : Auto  
Thumbnail : None  
Copyright :  
Exif comment :

#### #5 - 02/04/2018 12:06 AM - Stefan Hoffmeister

Apparent root cause: The file *location* on disk, containing "funky" (i.e. non-ASCII) characters.

For an image located in directory "C:\Users\stefan\Desktop\x" life is totally good for darktable; it also writes the metadata into the library database (select from above works).

For the same image located in directory "C:\Users\stefan\Documents\gdrive\xxxxxx\xxxx+\xxxx-xx-xx - xxxxxxxxxxxx - xxxxxxüxxxx" the failure is observed. Note that the xxx are masking true (ASCII) content, I left the ü in to trip any issues.

I consider the "+" and the "ü" candidates for trouble; the "ü" character, for being outside of ASCII, the "+" character possibly tripping directory splitting.

It is beyond my comprehension why, given that, sometimes darktable then even does show the camera model and maker, but never the rest, and never persists that into the database?

**#6 - 02/04/2018 12:20 AM - Stefan Hoffmeister**

Is it conceivable that

```
#if defined(_WIN32) && defined(EXV_UNICODE_PATH)
#define WIDEN pugi::as_wide(s)
#else
#define WIDEN (s)
```

at <https://github.com/darktable-org/darktable/blob/0b3259588fd986e0385bab021921ff5b55c3bbbed/src/common/exif.cc#L46> does not play well with whatever exiv2 expects?

Perhaps a different strategy can be used, e.g. as in

```
https://github.com/darktable-org/darktable/commit/dbccb8c2e69b4ba39d267b0b1fd5d910f08476c9#diff-b7cd9bf0c59dbd0fe62d17a5aecf1a62R177
```

? I have no idea what exiv2 expects, though.

**#7 - 02/04/2018 12:30 AM - Stefan Hoffmeister**

<https://github.com/Exiv2/exiv2/blob/55001c8dfff41390da1862234312f6486202204d/msvc/ReadMe.txt#L216> indicates that EXV\_UNICODE\_PATH must be explicitly enabled, and elsewhere <https://github.com/Exiv2/exiv2/blob/4beb08e2196a43ad88f55899eed72245e28d033c/include/exiv2/config.h#L161> indicates

```
#error EXV_UNICODE_PATH is not supported for MinGW builds
```

IIRC, darktable is built from a mingw build system(?), which would imply that WIDEN would never happen? What does the exiv2 library expect?

**#8 - 02/04/2018 12:34 AM - Stefan Hoffmeister**

<https://github.com/darktable-org/darktable/commit/c23bd4a5ad85eccab81681f6672c3846f21b978a> applied the WIDEN

**#9 - 02/04/2018 01:10 PM - Tobias Ellinghaus**

Using the 2.4.1 installer the file opens fine for me, with (more or less) the same path as you have. Could you please check if there are any error messages in the log file? <https://www.darktable.org/about/faq/#faq-windows-logs>

**#10 - 02/04/2018 02:31 PM - Stefan Hoffmeister**

There were no error messages present in the log file ("darktable.exe -d all"), but the log file contained references to some early (?) darktable build on Windows:

```
=====
version: darktable 2.3.0+866~gc23bd4a5a
start: 2017:11:05 17:11:14

end: 2017:11:05 17:11:27 =====
```

Given that, I decided to

- terminate darktable 2.4.1
- delete

- C:\Users\stefan\AppData\Local\darktable\data.db
- C:\Users\stefan\AppData\Local\darktable\library.db
- "darktable.exe -d all"

... et voilà, EXIF is present.

It would seem as if that early version of darktable on Windows placed bad EXIF information (well: **none**) into the cache (library.db). And 2.4.1 did not refresh this information at all (which proves yet again that caching is evil ;).

Would there have been a means to force re-reading EXIF?

FWIW, and only **very** tangentially related to this bug report because I ran into the FAQ:

- a) It would be helpful if <https://www.darktable.org/about/faq/#faq-windows-logs> referred to [https://www.darktable.org/usermanual/en/overview\\_chapter.html#darktable\\_commandline\\_parameters](https://www.darktable.org/usermanual/en/overview_chapter.html#darktable_commandline_parameters) and instructed the user to turn on logs
- b) <https://www.darktable.org/about/faq/#faq-windows-tiff> could be rephrased to include "Please upgrade to darktable 2.4.1 for much improved TIFF export performance." - see <https://redmine.darktable.org/issues/11884>

#### #11 - 02/04/2018 03:43 PM - Tobias Ellinghaus

- Status changed from New to Closed: invalid

Stefan Hoffmeister wrote:

There were no error messages present in the log file ("darktable.exe -d all"), but the log file contained references to some early (?) darktable build on Windows:

That is normal, new output is appended to the file.

version: darktable 2.3.0+866~gc23bd4a5a

That version had a bunch of bugs, not being able to read Exif data from those files was one of them.

Given that, I decided to

- terminate darktable 2.4.1
- delete
  - C:\Users\stefan\AppData\Local\darktable\data.db
  - C:\Users\stefan\AppData\Local\darktable\library.db

Just removing the images from darktable (select on lighttable, hit <del>) would have been enough.

- "darktable.exe -d all"

That is such a useless switch. In general there is no reason at all to use it. It will show a ton of garbage and hide anything that might be interesting.

... et voilà, EXIF is present.

It would seem as if that early version of darktable on Windows placed bad EXIF information (well: **none**) into the cache (library.db). And 2.4.1 did not refresh this information at all (which proves yet again that caching is evil ;).

Yes, darktable doesn't re-read files that were already imported.

Would there have been a means to force re-reading EXIF?

No.

FWIW, and only **very** tangentially related to this bug report because I ran into the FAQ:

a) It would be helpful if <https://www.darktable.org/about/faq/#faq-windows-logs> referred to [https://www.darktable.org/usermanual/en/overview\\_chapter.html#darktable\\_commandline\\_parameters](https://www.darktable.org/usermanual/en/overview_chapter.html#darktable_commandline_parameters) and instructed the user to turn on logs

If you mean the "-d all" you used then no, we will never advise any user to do that.

b) <https://www.darktable.org/about/faq/#faq-windows-tiff> could be rephrased to include "Please upgrade to darktable 2.4.1 for much improved TIFF export performance." - see <https://redmine.darktable.org/issues/11884>

Good idea.

#12 - 02/07/2018 09:30 PM - Roman Lebedev

- Target version set to 2.6.0

**Files**

---

P1010387.ORF	14.4 MB	02/03/2018	Stefan Hoffmeister
P1010387.ORF.EXIF.txt	10.4 KB	02/03/2018	Stefan Hoffmeister